

***IN THE CLAIMS***

Please amend the claims as follows:

1. (Previously Presented) A packet scheduler, comprising:  
inputs configured to receive requests for connecting multiple input ports to multiple output ports;  
outputs for configuring connections between the multiple input ports and the multiple output ports; and  
an arbitration circuit configured to conduct a first arbitration phase for a next time slot where multiple independent arbitrations are conducted concurrently for each of the multiple output ports, the arbitration circuit selecting the input ports for each of the output ports for the next time slot according to both a priority and weight of packets at the input ports used for the multiple independent arbitrations.

2. (Previously Presented) A scheduler according to claim 1 wherein the input ports that win multiple ones of the multiple arbitrations accept one of the arbitration wins and reject the other arbitration wins, the arbitration circuit then conducting another arbitration phase for the next time slot with independent arbitrations for each of the output ports that did not receive arbitration win acceptance from input ports during the first arbitration phase.

3. (Original) A scheduler according to claim 2 including timers that are activated anytime one of the input ports requests a connection to one of the output ports, the arbitration circuit increasing the priority for any input ports having unserved connection requests extending beyond a timer period.

4. (Original) A scheduler according to claim 1 wherein the arbitration circuit conducts output port arbitrations for each one of the output ports and conducts input port arbitrations for each one of the input ports winning multiple output port arbitrations.

5. (Original) A scheduler according to claim 4 wherein the output port arbitrations and the input port arbitrations are conducted for both multicast packets and unicast packets for a next time slot.

6. (Original) A scheduler according to claim 1 wherein each one of the input ports has associated virtual output queues each dedicated to a different one of the output ports.

7. (Original) A scheduler according to claim 6 wherein the arbitration circuit conducts output port arbitrations for all of the virtual output queues dedicated to the same output ports and conducts input port arbitrations between the virtual output queues for the same input port issued grants during the output port arbitrations.

8. (Original) A scheduler according to claim 7 wherein the output port arbitrations and the input port arbitrations are conducted according to both the priority and number of bytes of the packets associated with the virtual output queues.

9. (Original) A scheduler according to claim 8 including output port pointers for conducting round robin arbitrations during the output port arbitrations and input port pointers for conducting round robin arbitrations during the input port arbitrations.

10. (Original) A scheduler according to claim 9 including a cross switch that is configured by the arbitration circuit to connect the input ports to the output ports during individual time slots according to the output port arbitrations and the input port arbitrations.

11. (Original) A scheduler according to claim 1 wherein the arbitration circuit conducts a multicast arbitration that establishes connections for multicast packets during a next time slot and then conducts a unicast arbitration that establishes connections for unicast packets during the next time slot for any remaining unassigned output ports.

12. (Previously Presented) A scheduler according to claim 11 wherein the multicast arbitration and the unicast arbitration are conducted for both output port arbitrations and also for input port arbitrations.

13. (Currently Amended) A packet scheduler, comprising:  
inputs configured to receive requests for connecting input ports to output ports;  
outputs for configuring connections between the input ports and the output ports; and

an arbitration circuit configured to arbitrate between the input ports for connections to the output ports, the arbitration circuit selecting the input ports for a next time slot according to both a priority and weight of packets at the input ports,

wherein the arbitration circuit conducts a multicast arbitration that establishes connections for multicast packets during [[a]] the next time slot and then conducts a unicast arbitration that establishes connections for unicast packets during the next time slot for any remaining unassigned output ports, the multicast arbitration and the unicast arbitration conducted for both output port arbitrations and also for input port arbitrations and the arbitration circuit is programmable to vary a percentage of output ports assignable during the multicast arbitration.

14. (Canceled).

15. (Currently Amended) ~~A method according to claim 14~~ A method for scheduling connections between multiple input ports and multiple output ports, comprising:  
receiving requests from multiple input port buffers for connections to the multiple output ports during a next time slot;  
identifying arbitration parameters for the requests;  
conducting multiple independent output port arbitrations for each one of the multiple output ports according to the arbitration parameters;  
issuing grants to the input port buffers winning the output port arbitrations for each of the multiple output ports;  
conducting input port arbitrations for input ports receiving grants from multiple output port arbitrations;  
accepting one of the multiple grants for each of the input port arbitrations; and  
connecting the multiple output ports to the input port buffers accepting the grants;  
wherein the arbitration parameters include a weight that varies according to a number of packet bytes in the input port buffers and a priority of the packets in the input port buffers.

16. (Original) A method according to claim 15 including:  
selecting one of the weight and the priority to identify a first highest arbitration status;  
using the other nonselected weight and priority to identify a second highest arbitration status for two or more input port buffers having the same first highest arbitration status; and

issuing grants to the input port buffers according to the first highest arbitration status when packet for only one input port buffer has the first highest arbitration status and issuing grants to the input port buffers according to the second highest arbitration status when two or more input port buffers have the first highest arbitration status.

17. (Original) A method according to claim 16 including using a round robin arbitration  
when two or more input port buffers have the same first highest arbitration status and the same second highest arbitration status.

18. (Currently Amended) ~~A method according to claim 14~~ A method for scheduling connections between multiple input ports and multiple output ports, comprising:  
receiving requests from multiple input port buffers for connections to the multiple output ports during a next time slot;  
identifying arbitration parameters for the requests;  
conducting multiple independent output port arbitrations for each one of the multiple output ports according to the arbitration parameters;  
issuing grants to the input port buffers winning the output port arbitrations for each of the multiple output ports;  
conducting input port arbitrations for input ports receiving grants from multiple output port arbitrations;  
accepting one of the multiple grants for each of the input port arbitrations; including  
conducting second independent arbitrations between each of the output ports that did not receive a grant acceptance from one of the input ports and any of the input port buffers that did not previously receive grants; and  
connecting the multiple output ports to the input port buffers accepting the grants.

19. (Currently amended) ~~A method according to claim 14 including:~~ A method for scheduling connections between multiple input ports and multiple output ports, comprising:  
receiving requests from multiple input port buffers for connections to the multiple output ports during a next time slot;  
identifying arbitration parameters for the requests;  
conducting a multicast arbitration for multicast packets in the input port buffers;

[[and]]

conducting multiple independent output port arbitrations for each one of the multiple output ports according to the arbitration parameters;

granting one of the input port buffers winning the multicast arbitration all output ports identified in an associated multicast group vector;

issuing grants to the input port buffers winning the output port arbitrations for each of the multiple output ports;

conducting input port arbitrations for input ports receiving grants from multiple output port arbitrations;

accepting one of the multiple grants for each of the input port arbitrations; and

connecting the multiple output ports to the input port buffers accepting the grants.

20. (Currently amended) A method for scheduling connections between input ports and output ~~[[port]]~~ ports, comprising:

receiving requests from input port buffers for connections to the output ports during a next time slot;

identifying arbitration parameters for the requests;

conducting output port arbitrations for each one of the output ports according to the arbitration parameters;

issuing grants to the input port buffers winning the output port arbitrations;

conducting input port arbitrations for input ~~[[ports]]~~ port buffers receiving grants from the output port arbitrations;

accepting one of the grants to one of the input port buffers at each input port buffer winning at least one output ~~the input~~ port arbitration~~[[s]]~~;

connecting the input port buffers accepting the grants to the requested output ports;

conducting a multicast arbitration for multicast packets in the input port buffers;

granting one of the input port buffers winning the multicast arbitration all output ports identified in an associated multicast group vector;

identifying multicast group vectors for the multicast packets;

conducting multicast input port arbitrations for identifying a highest one of the multicast group vectors for each one of the input ~~[[ports]]~~ port buffers;

conducting multicast output port arbitrations using the highest multicast group vectors for the input ~~[[ports]]~~ port buffers; and

establishing connections for the multicast group vector winning the multicast output port arbitrations.

21. (Previously Presented) A method according to claim 20 including:  
comparing grants issued from the output port arbitrations with the multicast group vectors; and  
accepting the grants matching the multicast group vectors.

22. (Original) A method according to claim 19 including conducting a unicast arbitration after the multicast arbitration.

23. (Currently Amended) A method for scheduling connections between input ports and output [[port]] ports, comprising:  
receiving requests from input port buffers for connections to the output ports during a next time slot;  
identifying arbitration parameters for the requests;  
conducting output port arbitrations for each one of the output ports according to the arbitration parameters;  
issuing grants to the input port buffers winning the output port arbitrations;  
conducting input port arbitrations for input [[ports]] port buffers receiving grants from the output port arbitrations;  
accepting one of the grants to one of the input port buffers at each input port buffer winning at least one output ~~the input~~ port arbitration~~[[s]]~~;  
connecting the input port buffers accepting the grants to the requested output ports;  
conducting a multicast arbitration for multicast packets in the input port buffers;  
granting one of the input port buffers winning the multicast arbitration all output ports identified in an associated multicast group vector; and  
varying a percentage of output ports that can be assigned during the multicast arbitration before conducting ~~[[the]]~~ a unicast arbitration.

24. (Currently amended) A method according to claim ~~[[14]]~~ 15 including dedicating one input port buffer in each input port to one of the output ports and using the input port buffers ~~[[at]]~~ as virtual output buffers for temporarily storing packets assigned to the ~~dedicated~~ output ports to which the input port buffers are dedicated.

25. (Currently amended) A method according to claim [[14]] 15 including conducting a first round robin arbitration when multiple packets have a same highest priority and a same weight during the output port arbitrations and conducting a second round robin arbitration during the input port arbitrations when multiple input port buffers for the same input ports have been issued grants during the output port arbitration and have the same highest priority and the same weight.

26. (Canceled).

27. (Currently amended) ~~A network processing device according to claim 26~~ A network processing device, comprising:  
multiple input ports for receiving incoming packets;  
multiple output ports for outputting packets;  
a cross switch coupled to the input ports and the output ports;  
a scheduler that configures the cross switch for connecting multiple selected ones of the input ports to multiple selected ones of the output ports at the same time; and  
multiple virtual output buffers associated with each one of the input ports, each one of the virtual output buffers dedicated to a different one of the output ports, the scheduler conducting multiple independent output port arbitrations for each one of the multiple output ports and the associated dedicated virtual output buffers and configuring the cross-switch according to the multiple independent arbitrations, wherein the scheduler conducts a multicast arbitration before each time slot to select virtual output buffers to connect to multiple output ports, the scheduler then conducting a unicast arbitration for connecting any unselected virtual output buffers to unselected output ports.

28. (Previously Presented) A network processing device according to claim 27 wherein the scheduler conducts the multicast arbitration and the unicast arbitration both for the virtual output buffers associated with the same input ports and for the virtual output buffers assigned to the same output ports.

29. (Currently Amended) A network processing device according to claim 28 wherein the multicast arbitration and the unicast arbitration[[:]] :  
issue grants according to priority of the packets in the virtual output buffers;

issue grants according to a number of bytes in the packets when packets for two or more virtual output buffers have a same highest priority; and

issue grants according to a round robin order when packets for two or more virtual output buffers have the same highest priority and a same number of bytes.

30. (Previously Presented) A network processing device according to claim 29 wherein the multicast arbitration and the unicast arbitration determine connections between the input ports and the output ports before each time slot.

31. (Currently amended) ~~A network processing device according to claim 26~~ A network processing device, comprising:  
multiple input ports for receiving incoming packets;  
multiple output ports for outputting packets;  
a cross switch coupled to the input ports and the output ports;  
a scheduler that configures the cross switch for connecting multiple selected ones of the input ports to multiple selected ones of the output ports at the same time; and  
multiple virtual output buffers associated with each one of the input ports, each one of the virtual output buffers dedicated to a different one of the output ports, the scheduler conducting multiple independent output port arbitrations for each one of the multiple output ports and the associated dedicated virtual output buffers and configuring the cross-switch according to the multiple independent arbitrations, the scheduler including conducting a second separate set of independent arbitrations between any of the output ports that did not receive connection acceptance from an input port arbitration winner and the virtual output ports associated with the input ports that did not win one of the previous independent output port arbitrations.